Structure Preserving CCA Secure Encryption and Applications

Jan Camenisch[1], Kristiyan Haralambiev[2], Markulf Kohlweiss[3], Jorn Lapon[4], and Vincent Naessens[4]

[1] IBM Research Zürich, Switzerland
jca@zurich.ibm.com
[2] Computer Science Department, New York University, USA
kkh@cs.nyu.edu
[3] Microsoft Research Cambridge, UK
markulf@microsoft.com
[4] Katholieke Hogeschool Sint-Lieven, Ghent, Belgium
{jorn.lapon,vincent.naessens}@kahosl.be

Abstract. In this paper we present the first CCA-secure public key encryption scheme that is structure preserving, i.e., our encryption scheme uses only algebraic operations. In particular, it does not use hash-functions or interpret group elements as bit-strings. This makes our scheme a perfect building block for cryptographic protocols where parties for instance want to prove properties about ciphertexts to each other or to jointly compute ciphertexts. Our scheme is very efficient and is secure against adaptive chosen ciphertext attacks.

We also provide a few example protocols for which our scheme is useful. For instance, we present an efficient protocol for two parties, Alice and Bob, that allows them to jointly encrypt a given function of their respective secret inputs such that only Bob learns the resulting ciphertext, yet they are both ensured of the computation's correctness. This protocol serves as a building block for our second contribution which is a set of protocols that implement the concept of so-called oblivious trusted third parties. This concept has been proposed before, but no concrete realization was known.

Keywords: public-key encryption, structure preserving, oblivious trusted third party.

1 Introduction

Public key encryption and signature schemes have become indispensable building blocks for cryptographic protocols such as anonymous credential schemes, group signatures, anonymous voting schemes, and e-cash systems. In the design of such protocols, it is often necessary that one party be able to prove to another that it has correctly signed or encrypted a message without revealing the message and its signature or encryption. An efficient implementation of such proofs is possible if the signature and encryption schemes allows one to employ

D.H. Lee and X. Wang (Eds.): ASIACRYPT 2011, LNCS 7073, pp. 89–106, 2011.

generalized Schnorr [12] or Groth-Sahai proofs [21]. In the design of suitable signature and encryption schemes one should therefore stay within the realm of algebraic groups and not break the algebraic structures, for instance, by using hash-functions in an essential way.

When it comes to signature schemes, a designer can pick from a number of schemes that are suitable (e.g., [15,5,1]). For encryption schemes secure against adaptive chosen ciphertext attack (CCA) the situation is quite different. Two schemes that are somewhat suitable are the Camenisch-Shoup and the Cramer-Shoup encryption schemes [7,18], allowing for the verifiable encryption (and decryption) of discrete logarithms and group elements, respectively. Both these schemes make use of a cryptographic hash function to achieve security against chosen ciphertext attacks. These hash functions, unfortunately, prevent one from efficiently proving relations between the input and output of the encryption procedure. Such proofs, however, are an important feature in many advanced protocols. They are for instance required when two parties are to jointly encrypt (a function of) their respective inputs without revealing them or when a user is to prove knowledge of a ciphertext, e.g., as a part of a proof of knowledge of a leakage-resilient signature [22,20] (proving knowledge of a signature is a central tool in privacy-preserving protocols which so far is not possible for leakage-resilient signatures).

In this paper we present the first efficient *structure preserving* CCA secure encryption scheme. The term "structure-preserving" is borrowed from the notion of structure-preserving digital signatures [1]. An encryption scheme is called structure-preserving if its public keys, messages (plaintexts), and ciphertexts are group elements and the encryption and decryption algorithm consists only of group and pairing operations. We achieve structure preserving encryption by a novel implementation of the consistency check that ensures security against chosen ciphertext attacks. More precisely, we implement the consistency checks using a bilinear map between algebraic groups and embed all other ciphertext components in the pre-image group of that map. Our ciphertext consistency element(s) could be either one element in the target group or several group elements in the pre-image group. The former gives better efficiency, whereas the latter can be used in more scenarios, in particular those making use of Groth-Sahai proofs [21]. We prove our encryption scheme secure against chosen ciphertext attacks under the decisional linear assumption [6]. Our encryption scheme and protocols also support so-called labels [7] which are public messages attached to a ciphertext and are important in the scenario we consider in this paper to bind a decryption policy to the ciphertext.

Our new encryption scheme is well suited to build a variety of protocols. For instance, with our scheme the following protocol problems can be addressed which are common stumbling stones when designing advanced cryptographic protocols:

- Our scheme can be used in the construction of leakage-resilient signatures [20] which will then enable, for the first time, a user to efficiently prove knowledge of a leakage-resilient signature.

- A user, who is given a ciphertext and a Groth-Sahai proof that the ciphertext was correctly computed, is able to prove to a third party that it is in possession of such a ciphertext without revealing it.
- Two users can jointly compute a ciphertext (of a function) of two plaintexts such that neither party learns the plain text of the other party and only one of the parties learns the ciphertext.

The last problem typically appears in protocols that do some kind of conflict resolution via a trusted third party. Examples include anonymity lifting (revocation) in group signatures and in anonymous credential systems [14] and optimistic fair exchange [3]. In these scenarios, there are typically two parties, say Alice and Bob, who run a protocol with each other and then provide each other with ciphertexts that can in case of a mishap (such as abuse of anonymity, conflict, unfair abortion of the protocol, etc.) be presented to a third party for resolution by decryption. Hereby, it is of course important that (1) the trusted third party be involved in case of mishap only and (2) the parties can convince each other that the ciphertexts indeed contain the right information. Note that CCA security is crucial here, as the trusted third party effectively acts as a decryption oracle. So far, protocol designers have used verifiable encryption, which unfortunately has the disadvantage that both parties learn the ciphertext of the other party. Hence, Alice could for instance take Bob's ciphertext and bribe the TTP so that it would act normally for all decryption requests except when Bob's ciphertext is presented in which case the TTP would just ignore the request.

To address this problem Camenisch, Gross, and Heydt-Benjamin [10] propose the concept of *oblivious trusted third parties (OTP)*: here, such conflict resolution protocols are designed in such a way that the trusted third party is kept oblivious of the concrete instance of the conflict resolution protocol. This means if Bob goes to the TTP for resolution, he cannot possibly be discriminated as the TTP cannot tell whether it is contacted by Bob or some other person. Therefore, if the TTP would deny such requests too often, that would be known and so there is no reason for Bob to believe that the TTP will not resolve the conflict for him if need be. Unfortunately, Camenisch et al. only provide a high-level construction for such a protocol but do not present a concrete instantiation. Based on our new encryption scheme, we present the first concrete protocols that implement OTP.

We prove all our protocols secure under composable simulation-based security definitions [16,4,23].

Related Work. There is of course a lot of related work on encryption schemes, but our scheme is the first one that is structure preserving. Considering our second contribution, the protocols for oblivious trusted parties, the only related work is by Camenisch, Gross, and Heydt-Benjamin [10]. They introduced the concept of oblivious trusted third parties but, as we mentioned, do not provide any concrete protocol.

2 Structure Preserving Encryption

In this section, we define the notion of structure-preserving encryption and present the first instantiation of such a scheme. The term "structure-preserving" is borrowed from the notion of structure-preserving digital signatures [1], and, for encryption, represents the idea that ciphertexts are constructed purely using (bilinear) group operations.

Note that the well known Cramer-Shoup [17,18] and Camenisch-Shoup [7] encryption schemes are not structure preserving as they make use of a cryptographic hash function. Even the hash-free variant of Cramer-Shoup is not structure preserving; that is because its consistency check requires group elements to be interpreted as exponents, which is not a group operation. The details of a proof of knowledge of a hash-free ciphertext would depend on the group's internal structure, e.g., it might be based on so called double-discrete logarithm proofs [8], which are bit-wise and thus much less efficient than standard discrete logarithm representation proofs.

Definition 1. Structure Preserving Encryption. *An encryption scheme is said to be structure-preserving if (1) its public keys, messages, and ciphertexts consist entirely of elements of a bilinear group, (2) its encryption and decryption algorithm perform only group and bilinear map operations, and (3) it is provably secure against chosen-ciphertext attacks.*

2.1 Basic Notation

We work in a group $\mathbb{G}$ of prime order q generated by g and equipped with a non-degenerate efficiently computable bilinear map $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Also, recall the well-known DLIN assumption [6]:

Definition 2. *Decisional Linear Assumption (DLIN). Let $\mathbb{G}$ be a group of prime order q. For randomly chosen $g_1, g_2, g_3 \leftarrow \mathbb{G}$ and $r, s, t \leftarrow \mathbb{Z}_q$, the following two distributions are computationally indistinguishable:*

$$(\mathbb{G}, g_1, g_2, g_3, g_1^r, g_2^s, g_3^t) \approx (\mathbb{G}, g_1, g_2, g_3, g_1^r, g_2^s, g_3^{r+s}) \ .$$

2.2 Construction

We construct a structure-preserving encryption scheme secure under DLIN. The scheme shares some similarities with the Cramer-Shoup encryption and with the Linear Cramer-Shoup encryption described by Shacham [24], neither of which is structure-preserving (even for their hash-free variants).

For simplicity, we describe the scheme when encrypting a message that is a single group element in $\mathbb{G}$, but it is easily extended to encrypt vectors of group elements. The extension is presented in the full version of the paper. Also, our scheme supports labels. We consider the case when a label L is a single group element, but the scheme extends trivially for the case of a label which is a vector of group elements. Labels from the space $\{0,1\}^*$ could be hashed to one or several

group elements, though in such cases they have to be part of the statement rather than the witness for any NIZK proof.

- $\mathsf{KeyGen}(1^\lambda)$: Choose random group generators $g_1, g_2, g_3 \leftarrow \mathbb{G}^*$. For randomly chosen $\boldsymbol{\alpha} \leftarrow \mathbb{Z}_q^3$, set $h_1 = g_1^{\alpha_1} g_3^{\alpha_3}$ and $h_2 = g_2^{\alpha_2} g_3^{\alpha_3}$. Then, select $\boldsymbol{\beta}_0, \ldots, \boldsymbol{\beta}_5 \leftarrow \mathbb{Z}_q^3$, and compute $f_{i,1} = g_1^{\beta_{i,1}} g_3^{\beta_{i,3}}$, $f_{i,2} = g_2^{\beta_{i,2}} g_3^{\beta_{i,3}}$, for $i = 0, \ldots, 5$. Output $\mathsf{pk} = (g_1, g_2, g_3, h_1, h_2, \{f_{i,1}, f_{i,2}\}_{i=0}^5)$ and $\mathsf{sk} = (\boldsymbol{\alpha}, \{\boldsymbol{\beta}_i\}_{i=0}^5)$.
- $\mathsf{Enc}(\mathsf{pk}, L, m)$: To encrypt a message m with a label L, choose random $r, s \leftarrow \mathbb{Z}_q$ and set

$$u_1 = g_1^r,\ u_2 = g_2^s,\ u_3 = g_3^{r+s}, \quad c = m \cdot h_1^r h_2^s,$$

$$v = \prod_{i=0}^{3} \hat{e}(f_{i,1}^r f_{i,2}^s, u_i) \cdot \hat{e}(f_{4,1}^r f_{4,2}^s, c) \cdot \hat{e}(f_{5,1}^r f_{5,2}^s, L),$$

  where $u_0 = g$. Output $\mathsf{c} = (u_1, u_2, u_3, c, v)$.
- $\mathsf{Dec}(sk, L, \mathsf{c})$: Parse c as (u_1, u_2, u_3, c, v). Then check whether

$$v \stackrel{?}{=} \prod_{i=0}^{3} \hat{e}(u_1^{\beta_{i,1}} u_2^{\beta_{i,2}} u_3^{\beta_{i,3}}, u_i) \cdot \hat{e}(u_1^{\beta_{4,1}} u_2^{\beta_{4,2}} u_3^{\beta_{4,3}}, c) \cdot \hat{e}(u_1^{\beta_{5,1}} u_2^{\beta_{5,2}} u_3^{\beta_{5,3}}, L),$$

  where $u_0 = g$. If the latter is unsuccessful, reject the ciphertext as invalid. Otherwise, output $m = c \cdot (u_1^{\alpha_1} u_2^{\alpha_2} u_3^{\alpha_3})^{-1}$.

Note that the ciphertext $\mathsf{c} \in \mathbb{G}^4 \times \mathbb{G}_T$. Using the pairing randomization techniques of [2], $v \in \mathbb{G}_T$ can be replaced by six random group elements $v_0, \ldots, v_5 \in \mathbb{G}$ for which the following equation holds: $v = \prod_{i=0}^{3} \hat{e}(v_i, u_i) \cdot \hat{e}(v_4, c) \cdot \hat{e}(v_5, L)$. This way, the ciphertext would consist only of elements in $\mathbb{G}$. The modification is straightforward and is described in the full version of this paper [11].

2.3 Correctness and Security

To observe the correctness of the decryption, note that

$$\begin{aligned} c \cdot (u_1^{\alpha_1} u_2^{\alpha_2} u_3^{\alpha_3})^{-1} &= m \cdot h_1^r h_2^s \cdot \left((g_1^r)^{\alpha_1} (g_2^s)^{\alpha_2} (g_3^{r+s})^{\alpha_3}\right)^{-1} \\ &= m \cdot (g_1^{\alpha_1} g_3^{\alpha_3})^r (g_2^{\alpha_2} g_3^{\alpha_3})^s \cdot \left((g_1^r)^{\alpha_1} (g_2^s)^{\alpha_2} (g_3^{r+s})^{\alpha_3}\right)^{-1} = m. \end{aligned}$$

The correctness of the validity element v can be verified similarly.

Next, we show the CCA security of the encryption scheme. Our security proof follows the high level idea of the Hash Proof System (HPS) paradigm [19]. Essentially, Lemma 1 says the "proof" π, which is used as a one-time pad for the encryption of the message, has a corresponding HPS which is 1-universal, whereas Lemma 2 shows that the "proof" φ, which constitutes the consistency check element, has a corresponding HPS that is 2-universal. To make the proof below more accessible to readers unfamiliar with the HPS paradigm, we opt for a self-contained proof which can be easily translated into the HPS framework.

Theorem 1. *If DLIN holds, the above public key encryption scheme is secure against chosen-ciphertext attacks (CCA).*

Proof sketch of Theorem 1: We proceed in a sequence of games. We start with a game where the challenger behaves like in the standard IND-CCA game (i.e., the challenge ciphertext is an encryption of m_b, for a randomly chosen bit b, where m_0, m_1 are messages given by the adversary), and end up with a game where the challenge ciphertext is an encryption of a message chosen uniformly at random from the message space. Then we show that all those games are computationally indistinguishable. Let W_i denote the event that the adversary $\mathcal{A}$ outputs b' such that $b = b'$ in Game i.

Game 0. This is the standard IND-CCA game. $Pr[W_0] = \frac{1}{2} + \mathsf{Adv}^{\mathcal{A}}(\lambda)$.

Game 1. For (m_0, m_1, L) chosen by the adversary, the challenge ciphertext $\mathfrak{c} = (\boldsymbol{u}, c, v)$ is computed using the "decryption procedure", i.e., $u_1 = g_1^r$, $u_2 = g_2^s$, $u_3 = g_3^{r+s}$, $c = m_b \cdot u_1^{\alpha_1} u_2^{\alpha_2} u_3^{\alpha_3}$ and $v = \prod_{i=0}^{3} \hat{e}(u_1^{\beta_{i,1}} u_2^{\beta_{i,2}} u_3^{\beta_{i,3}}, u_i) \cdot \hat{e}(u_1^{\beta_{4,1}} u_2^{\beta_{4,2}} u_3^{\beta_{4,3}}, c) \cdot \hat{e}(u_1^{\beta_{5,1}} u_2^{\beta_{5,2}} u_3^{\beta_{5,3}}, L)$. The change is only syntactical, so the two games produce the same distributions. $Pr[W_1] = Pr[W_0]$.

Game 2. The randomness vector $\boldsymbol{u} = (u_1, u_2, u_3)$ of the challenge ciphertext is computed as non-DLIN tuple, i.e., $u_1 = g_1^r$, $u_2 = g_2^s$, $u_3 = g_3^t$ where $r, s, t \leftarrow \mathbb{Z}_q$ and $r + s \neq t$. Game 1 and Game 2 are indistinguishable by DLIN. Therefore, $\mid Pr[W_2] - Pr[W_1] \mid = \mathrm{negl}(\lambda)$.

Game 3. First note that in the previous game, as well as in this one, any decryption query with "correct" ciphertext, i.e., which has a randomness vector a DLIN tuple, yields a unique plaintext. That is, regardless of the concrete choice of sk which matches pk seen by the adversary, such queries do not reveal any information about the secret key.

In this game, unlike the previous one, any decryption query with "malformed" ciphertext, i.e, which has a non-DLIN randomness vector $\widehat{\boldsymbol{u}}$, is rejected. Let's consider two cases:

- $(\widehat{\boldsymbol{u}}, \widehat{c}, \widehat{L}) = (\boldsymbol{u}, c, L)$. Such decryption query is rejected because it is either the challenge ciphertext (when $\widehat{v} = v$) or the verification predicate fails trivially (when $\widehat{v} \neq v$). So, this case is the same in Game 2 and Game 3.
- $(\widehat{\boldsymbol{u}}, \widehat{c}, \widehat{L}) \neq (\boldsymbol{u}, c, L)$. By Lemma 2, such decryption query is rejected in Game 2 with overwhelming probability, whereas in Game 3 it is always rejected.

As the number of decryption queries is polynomial, $\mid Pr[W_3] - Pr[W_2] \mid = \mathrm{negl}(\lambda)$.

Game 4. The challenge ciphertext encrypts a random message from the message space. Game 3 and Game 4 are (information theoretically) indistinguishable by Lemma 1. $Pr[W_4] = Pr[W_3]$.

In the last game, the challenger's choice b is independent from the ciphertext, so $Pr[W_4] = \frac{1}{2}$. Then, by the indistinguishability of the consecutive games $Pr[W_0] = \frac{1}{2} + \mathrm{negl}(\lambda)$, hence $\mathsf{Adv}^{\mathcal{A}}(\lambda) = \mathrm{negl}(\lambda)$. □

Lemma 1 which we used in the above proof says that the one-time pad of the message, when computing the challenge ciphertext in Game 4, can be replaced by a random element. Whereas Lemma 2 shows that any decryption query with "malformed" ciphertext $\widehat{\mathfrak{c}}$ is rejected with overwhelming probability because the adversary $\mathcal{A}$ can hardly do better than guess the correct validity element.

For the formulation and proof of the lemmas, let $g_1, g_2, g_3 \leftarrow \mathbb{G}^*$ and $u_1 = g_1^r$, $u_2 = g_2^s$, $u_3 = g_3^t$, where r, s, t are randomly chosen from $\mathbb{Z}_q$ and $r + s \neq t$. And for convenience, denote $z_1 = \mathrm{dlog}_g(g_1)$, $z_2 = \mathrm{dlog}_g(g_2)$, and $z_3 = \mathrm{dlog}_g(g_3)$.

Lemma 1. *For randomly chosen $\boldsymbol{\alpha} \leftarrow \mathbb{Z}_q^3$, let $h_1 = g_1^{\alpha_1} g_3^{\alpha_3}$, $h_2 = g_2^{\alpha_2} g_3^{\alpha_3}$, and $\pi = u_1^{\alpha_1} u_2^{\alpha_2} u_3^{\alpha_3}$. Then, for a randomly chosen $\psi \leftarrow \mathbb{G}$ it is true that the following distributions are equivalent: $(h_1, h_2, \pi) \equiv (h_1, h_2, \psi)$.*

Proof sketch of Lemma 1: Note that $h_1 = g^{\alpha_1 z_1 + \alpha_3 z_3}$ and $h_2 = g^{\alpha_2 z_2 + \alpha_3 z_3}$. Then, for the tuple (h_1, h_2, π) the following equation holds:

$$\begin{pmatrix} z_1 & 0 & z_3 \\ 0 & z_2 & z_3 \\ rz_1 & sz_2 & tz_3 \end{pmatrix} \cdot \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} = \begin{pmatrix} \mathrm{dlog}_g(h_1) \\ \mathrm{dlog}_g(h_2) \\ \mathrm{dlog}_g(\pi) \end{pmatrix}$$

Denote the matrix with M. It has a determinant $\det(M) = z_1 z_2 z_3 (t - r - s)$ which is not equal to 0 due to the choice of the parameters. Therefore the matrix is invertible, and for any $\pi \in \mathbb{G}$, and fixed h_1, h_2, there exists a unique $\boldsymbol{x}$ which yields the tuple (h_1, h_2, π). □

Lemma 2. *Let $\widehat{\boldsymbol{u}} = (\widehat{u}_1, \widehat{u}_2, \widehat{u}_3)$ be any tuple such that $\widehat{u}_1 = g_1^{\widehat{r}}$, $\widehat{u}_2 = g_2^{\widehat{s}}$, and $\widehat{u}_3 = g_3^{\widehat{t}}$, for $\widehat{r} + \widehat{s} \neq \widehat{t}$. And for randomly chosen $\boldsymbol{\beta}_0, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_5 \leftarrow \mathbb{Z}_q^3$, let $f_{i,1} = g_1^{\beta_{i,1}} g_3^{\beta_{i,3}}$, $f_{i,2} = g_2^{\beta_{i,2}} g_3^{\beta_{i,3}}$, for $i = 0, \ldots, 5$. For any $\boldsymbol{m}$ and $\widehat{\boldsymbol{m}}$ in $\mathbb{G}^5$, let*

$$\varphi = \prod_{i=0}^{5} \hat{e}(u_1^{\beta_{i,1}} u_2^{\beta_{i,2}} u_3^{\beta_{i,3}}, m_i) \quad \text{and} \quad \widehat{\varphi} = \prod_{i=0}^{5} \hat{e}((\widehat{u}_1)^{\beta_{i,1}} (\widehat{u}_2)^{\beta_{i,2}} (\widehat{u}_3)^{\beta_{i,3}}, \widehat{m}_i),$$

where $m_0 = \widehat{m}_0 = g$. Then, for any $\boldsymbol{m}$ and $\widehat{\boldsymbol{m}}$, $\boldsymbol{m} \neq \widehat{\boldsymbol{m}}$, it is true that the following two distributions are equivalent: $(\{f_{i,1} f_{i,2}\}_{i=0}^5, \varphi, \widehat{\varphi}) \equiv (\{f_{i,1} f_{i,2}\}_{i=0}^5, \varphi, \psi)$, where $\psi \leftarrow \mathbb{G}_T$ is randomly chosen.

Proof sketch of Lemma 2: Similarly to the proof of the previous lemma, let's define all variables which depend on $\{\boldsymbol{\beta}_i\}_{i=0}^5$ as the result of a constant matrix M multiplied by the vector $(\boldsymbol{\beta}_0^\top || \boldsymbol{\beta}_1^\top || \ldots || \boldsymbol{\beta}_5^\top)^\top$. For convenience, denote with $w_i = \mathrm{dlog}_g(m_i)$ and $\widehat{w}_i = \mathrm{dlog}_g(\widehat{m}_i)$, for $i = 1, \ldots, 5$. Then, we have:

$$\begin{pmatrix} z_1 & 0 & z_3 & - & - & - & \dots & - & - & - \\ 0 & z_2 & z_3 & - & - & - & \dots & - & - & - \\ - & - & - & z_1 & 0 & z_3 & \dots & - & - & - \\ - & - & - & 0 & z_2 & z_3 & \dots & - & - & - \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ - & - & - & - & - & - & \dots & z_1 & 0 & z_3 \\ - & - & - & - & - & - & \dots & 0 & z_2 & z_3 \\ rz_1 & sz_2 & tz_3 & w_1rz_1 & w_1sz_2 & w_1tz_3 & \dots & w_5rz_1 & w_5sz_2 & w_5tz_3 \\ \widehat{r}z_1 & \widehat{s}z_2 & \widehat{t}z_3 & \widehat{w}_1\widehat{r}z_1 & \widehat{w}_1\widehat{s}z_2 & \widehat{w}_1\widehat{t}z_3 & \dots & \widehat{w}_5\widehat{r}z_1 & \widehat{w}_5\widehat{s}z_2 & \widehat{w}_5\widehat{t}z_3 \end{pmatrix} \cdot \begin{pmatrix} | \\ \boldsymbol{\beta}_0 \\ | \\ \vdots \\ | \\ \boldsymbol{\beta}_5 \\ | \end{pmatrix} = \begin{pmatrix} \mathrm{dlog}_g(f_{0,1}) \\ \mathrm{dlog}_g(f_{0,2}) \\ \mathrm{dlog}_g(f_{1,1}) \\ \mathrm{dlog}_g(f_{2,2}) \\ \vdots \\ \mathrm{dlog}_g(f_{5,1}) \\ \mathrm{dlog}_g(f_{5,2}) \\ \mathrm{dlog}(\varphi) \\ \mathrm{dlog}(\widehat{\varphi}) \end{pmatrix}.$$

We would like to argue that the rows of the matrix M are linearly independent. As there exists i, $i \geq 1$, such that $m_i \neq \widehat{m}_i$, if we choose the sub-matrix M' consisting of the intersection of the last two rows and rows $1,\ 2,\ 2i+1,\ 2i+2$ with columns $1,\ 2,\ 3,\ 3i+1,\ 3i+2,\ 3i+3$, we get:

$$M' = \begin{pmatrix} z_1 & 0 & z_3 & 0 & 0 & 0 \\ 0 & z_2 & z_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & z_1 & 0 & z_3 \\ 0 & 0 & 0 & 0 & z_2 & z_3 \\ rz_1 & sz_2 & tz_3 & w_irz_1 & w_isz_2 & w_itz_3 \\ \widehat{r}z_1 & \widehat{s}z_2 & \widehat{t}z_3 & \widehat{w}_i\widehat{r}z_1 & \widehat{w}_i\widehat{s}z_2 & \widehat{w}_i\widehat{t}z_3 \end{pmatrix}.$$

If the rows of M are not linearly independent, so are the rows of M'. However, M' has a determinant $\det(M') = \pm z_1^2 z_2^2 z_3^2 (w_i - \widehat{w}_i)(t - r - s)(\widehat{t} - \widehat{r} - \widehat{s})$ which is not equal to 0 due to choice of the parameters. Therefore, the rows of M are linearly independent. □

3 Secure Joint Ciphertext Computation

The CCA secure structure preserving encryptions scheme is well suited to build a variety of protocols. More specifically, it facilitates the construction of protocols that make use of practical ZK protocols to prove properties about partial ciphertexts. We consider a two-party protocol for the joint computation of a ciphertext under a third-party public key pk. The encrypted value is a function of two secrets, each of which remains secret from the other protocol participant. Moreover, only one participant gets to know the ciphertext. We study the case where only the first party learns the ciphertext whereas the second one has no output.

3.1 Preliminaries

Simulatability Model. We use strong simulation-based definitions that guarantee security under composition in the flavor of [16,4,23]. In particular we base our exposition on [23]. In [23] both ideal systems $\mathcal{I}$ and their realizations as

cryptographic protocols $\mathcal{P}$ are configurations of multi-tape interactive Turing machines (ITMs). An ITM is triggered by another ITM if the latter writes a message on an output tape that corresponds to an input tape of the former. As a convention we bundle communication tapes into interfaces inf where an interface consists of named input/output tape pairs. An input/output tape pair is named $\mathsf{inf.R}$ after a combination of the interface name inf and a role name R. We refer to the set of all roles of an interface as $\mathsf{inf}.\mathcal{R}$.

For simulation-based security definitions, the ideal system $\mathcal{I}$ and the protocol $\mathcal{P}$ that emulates this ideal system, have to present the same interface inf towards their environment, i.e., they must be *environment compatible*. We refer to an ideal system and a protocol that is environment compatible with respect to interface inf as $\mathcal{I}_{\mathsf{inf}}$ and $\mathcal{P}_{\mathsf{inf}}$, respectively. In addition $\mathcal{I}_{\mathsf{inf}}$ and $\mathcal{P}_{\mathsf{inf}}$ expose different network interfaces, the simulator interface $\mathsf{inf}_{\mathsf{Sim}}$ and the adversary interface $\mathsf{inf}_{\mathsf{Adv}}$, respectively.

Strong simulatability. A proof that $\mathcal{P}_{\mathsf{inf}}$ emulates $\mathcal{I}_{\mathsf{inf}}$, short $\mathcal{P}_{\mathsf{inf}} \leq^{SS} \mathcal{I}_{\mathsf{inf}}$ will need to prove existence of a simulator Sim that translates between the interfaces $\mathsf{inf}_{\mathsf{Sim}}$ and $\mathsf{inf}_{\mathsf{Adv}}$ such that for all p.p.t. Env: $\mathsf{Env}|\mathcal{P}_{\mathsf{inf}} \approx \mathsf{Env}|\mathsf{Sim}|\mathcal{I}_{\mathsf{inf}}$. This is formalized as *strong simulatability* which implies other simulatability notions such as universal composability with dummy adversaries and blackbox simulatability.

Corruption. We consider only static corruption. A corrupted role in the ideal and in the real world is controlled through $\mathsf{inf}_{\mathsf{Sim}}.\mathsf{R}$ and $\mathsf{inf}_{\mathsf{Adv}}.\mathsf{R}$ respectively, and acts as a proxy that allows the simulator, respectively, the environment to send messages to any of its other connected tapes. We consider ideal systems $\mathcal{I}_{\mathsf{inf}}$ that are fully described by a virtual incorruptible party $\mathcal{F}_{\mathsf{inf}}$. As the functionality $\mathcal{F}_{\mathsf{inf}}$ implements the security critical parts of an ideal system, the ITM's representing the different roles of the interface only need to implement forwarding and corruption. We refer to the dummy party of role R as $\mathcal{D}_{\mathsf{R}}$. When operating over an adversarially controlled network, even an ideal cryptographic system cannot prevent denial of service attacks. We therefor give the adversary the possibility to delay messages from the ideal functionality to dummies.

Practical Zero-Knowledge Proof of Knowledge Protocols. For the types of relations required in our protocols, there exist practical ZK protocols. We refer to Camenisch et al. [9,13] for details. We will be proving statements of the form $\mathsf{ZK}\, w_1, \ldots, w_n : \phi(w_1, \ldots, w_n, \mathit{bases})$ where w_i are exponents and ϕ is a predicate defining discrete logarithm representations. For a more detailed description, we refer to the full version of this paper.

We use a zero-knowledge ideal functionality as defined by Listing 1 that is a simplification of the $\mathcal{F}_{ZK}^{R,R'}$ functionality of [9] for which we consider only static corruption. This allows us to reuse their ZK protocol compiler to obtain efficient multi-session instantiations $\mathcal{P}_{\mathsf{zk}}$ of $\mathcal{I}_{\mathsf{zk}}(\mathfrak{R})$ in the hybrid secure channel and joint-state common reference string model.

Listing 1: Functionality $\mathcal{F}_{\mathsf{zk}}(\mathfrak{R})$:

$\mathcal{F}_{\mathsf{zk}}$ *receives input from* $\mathcal{D}_{\mathsf{Pv}}$ *over* $\mathcal{F}_{\mathsf{zk}}.\mathsf{Pv}$ *and provides output to* $\mathcal{D}_{\mathsf{Vf}}$ *through the delayed communication tape* $\mathcal{F}_{\mathsf{zk}}.\mathsf{Vf}$. *Variable state is initialized to "ready".*

On $(\mathtt{Prove}, inst, wit)$ *from* $\mathcal{F}_{\mathsf{zk}}.\mathsf{Pv}$ *where state = "ready" and* $(inst, wit) \in \mathfrak{R}$

– *let state = "final"; send* $(\mathtt{Prove}, inst)$ *to* $\mathcal{F}_{\mathsf{zk}}.\mathsf{Vf}$

Two-party computation. In conformance with the simulatability model discussed above, Listing 2 defines the ideal functionality for the joint computation of any function f on verifiable inputs inp_1 and inp_2. When performing such a two-party computation, party P_{1+i} is guaranteed that P_{2-i} knows a witness wit_{2-i} for its input inp_{2-i} such that $(inst, (wit_{2-i}, inp_{2-i})) \in \mathfrak{R}_{2-i}$. We restrict ourselves to tractable relations $\mathfrak{R}_i$ for which we can give efficient universally composable proofs of knowledge as described in the full paper.

Listing 2: Functionality $\mathcal{F}_{\mathsf{tpc}}(f, \mathfrak{R}_1, \mathfrak{R}_2)$

$\mathcal{F}_{\mathsf{tpc}}$ *communicates with* $\mathcal{D}_{\mathsf{P}_1}$ *and* $\mathcal{D}_{\mathsf{P}_2}$ *through delayed communication tapes* $\mathcal{F}_{\mathsf{tpc}}.\mathsf{P}_1$ *and* $\mathcal{F}_{\mathsf{tpc}}.\mathsf{P}_2$. *Variables inst, pub,* inp_1 *store the input of the first party; variable state is initialized to "ready".*

On $(\mathtt{Input}_1, inst', pub', wit_1', inp_1')$ *from* $\mathcal{F}_{\mathsf{tpc}}.\mathsf{P}_1$ *where state = "ready" and* $(inst', (wit_1', inp_1')) \in \mathfrak{R}_1$

– *let* $inp_1 = inp_1'$, $inst = inst'$, $pub = pub'$, *and state = "input1"; send* $(\mathtt{Input}_1, inst, pub)$ *to* $\mathcal{F}_{\mathsf{tpc}}.\mathsf{P}_2$

On $(\mathtt{Input}_2, wit_2, inp_2)$ *from* $\mathcal{F}_{\mathsf{tpc}}.\mathsf{P}_2$ *where state = "input1" and* $(inst, (wit_2, inp_2)) \in \mathfrak{R}_2$

– *let state = "final"; send* $(\mathtt{Result}, f(pub, inp_1, inp_2))$ *to* $\mathcal{F}_{\mathsf{tpc}}.\mathsf{P}_1$

We model an ideal secure two-party computation system $\mathcal{I}_{\mathsf{tpc}}(f, \mathfrak{R}_1, \mathfrak{R}_2)$ with interface tpc as the combination of two dummy Parties $\mathcal{D}_{\mathsf{P}_1}$ and $\mathcal{D}_{\mathsf{P}_2}$ and an ideal two party computation functionality $\mathcal{F}_{\mathsf{tpc}}$.

3.2 Construction

Model. The model of our joint ciphertext computation, is fully described by a secure two party computation as in Listing 2, where $inp_i = (l_i, \boldsymbol{x_i})$, $pub = \mathsf{pk}$, and f is $f_{\mathsf{JC}}(\mathsf{pk},\ (l_1, \boldsymbol{x}_1),\ (l_2, \boldsymbol{x}_2)\ ) = \mathsf{Enc}(\mathsf{pk}, g^{l_1+l_2}, (g^{x_{1,1}+x_{1,2}}, \ldots, g^{x_{n,1}+x_{n,2}}))$.

Implementation. We present the protocol for the special case where the jointly computed ciphertext encrypts a single message (i.e., $n = 1$). This extends trivially in the multi-message case.

The idea of the protocol is as follows. The first party computes a partial and blinded encryption of her secret, she proves that the computation is carried out correctly, and sends the partial encryption to the other party. The second party

takes the values from the first flow of the protocol and, using its secret and some randomness, computes a blinded full encryption of the agreed function of the two plaintext contributions. Then, the second party sends these values and proves that they are computed correctly. Finally, the first party unblinds the ciphertext and updates the consistency element to obtain a valid encryption of the function of the two secrets under jointly chosen randomness. The function can be a constant to the power of any polynomial of the two secrets; for simplicity, we consider the function $g^{x_1+x_2}$ where g is a fixed group element and x_1, x_2 are the two secrets.

Listing 3: Protocol $\mathcal{P}_{\mathsf{jcc}}(\mathfrak{R}_1, \mathfrak{R}_2)$

Party P_1 *and* P_2 *receive input from* $\mathsf{jcc.P}_1$ *and* $\mathsf{jcc.P}_2$ *respectively and communicate over* $\mathcal{I}_{\mathsf{zk}_1}$ *and* $\mathcal{I}_{\mathsf{zk}_2}$.

On $(\mathtt{Input}_1, inst, \mathsf{pk}, wit_1, (l_1, x_1))$ *from* $\mathsf{jcc.P}_1$

- *if* $(inst, (wit_1, l_1, x_1)) \notin \mathfrak{R}_1$, P_1 *aborts*
- P_1 *computes* $(msg_1, aux_1) \leftarrow \mathsf{BlindEnc}_1(\mathsf{pk}, l_1, x_1)$ *and proves* $((msg_1, \mathsf{pk}, inst), (wit_1, l_1, x_1, aux_1)) \in \mathfrak{R}_{\mathsf{P}_1}(\mathfrak{R}_1)$ *to* P_2 *using* $\mathcal{I}_{\mathsf{zk}_1}(\mathfrak{R}_{\mathsf{P}_1}(\mathfrak{R}_1))$
- P_2 *learns* $(msg_1, \mathsf{pk}, inst)$ *from* $\mathcal{I}_{\mathsf{zk}_1}$ *and outputs* $(\mathtt{Input}_1, inst, \mathsf{pk})$ *to* $\mathsf{jcc.P}_2$

On $(\mathtt{Input}_2, wit_2, (l_2, x_2))$ *from* $\mathsf{jcc.P}_2$

- *if* $(inst, (wit_2, l_2, x_2)) \notin \mathfrak{R}_2$, P_2 *aborts*
- P_2 *runs* $(msg_2, aux_2) \leftarrow \mathsf{BlindEnc}_2(\mathsf{pk}, l_2, x_2, msg_1)$
- P_2 *proves* $((msg_2, \mathsf{pk}, inst), (wit_2, l_2, x_2, aux_2)) \in \mathfrak{R}_{\mathsf{P}_2}(\mathfrak{R}_2)$ *to* P_1 *using* $\mathcal{I}_{\mathsf{zk}_2}(\mathfrak{R}_{\mathsf{P}_2}(\mathfrak{R}_2))$
- P_1 *learns* $(msg_2, \mathsf{pk}, inst)$ *from* $\mathcal{I}_{\mathsf{zk}_2}$, *computes* $\mathsf{c} \leftarrow \mathsf{UnblindEnc}(\mathsf{pk}, msg_2, aux_1)$, *and outputs* $(\mathtt{Result}, \mathsf{c})$ *to* $\mathsf{jcc.P}_1$

Where abstractly, relations $\mathfrak{R}_{\mathsf{P}_1}(\mathfrak{R}_1)$ and $\mathfrak{R}_{\mathsf{P}_2}(\mathfrak{R}_2)$ are defined as

$$\mathfrak{R}_{\mathsf{P}_1}(\mathfrak{R}_1) = \{(msg_1, \mathsf{pk}, inst), (wit_1, l_1, x_1, aux_1)) \mid \exists r : (msg_1, aux_1) = \mathsf{BlindEnc}_1(\mathsf{pk}, l_1, x_1; r) \wedge (inst, (wit_1, l_1, x_1)) \in \mathfrak{R}_1\}$$
$$\mathfrak{R}_{\mathsf{P}_2}(\mathfrak{R}_2) = \{((msg_2, \mathsf{pk}, inst), (wit_2, l_2, x_2, aux_2)) \mid \exists r : (msg_2, aux_2) = \mathsf{BlindEnc}_2(\mathsf{pk}, l_2, x_2, msg_1; r) \wedge (wit_2, l_2, x_2)) \in \mathfrak{R}_2\} .$$

In the full paper, we show how to efficiently prove the relations $\mathfrak{R}_{\mathsf{P}_1}(\mathfrak{R}_1))$ and $\mathfrak{R}_{\mathsf{P}_2}(\mathfrak{R}_2))$ by giving a $\mathcal{K}$ language statement.

We now give the details for the $\mathsf{BlindEnc}_1$, $\mathsf{BlindEnc}_2$, and $\mathsf{UnblindEnc}$ algorithms.

Listing 4: Algorithms of $\mathcal{P}_{\mathsf{jcc}}$

$(msg_1, aux_1) \leftarrow \mathsf{BlindEnc_1}(\mathsf{pk}, l_1, x_1)$

- *parse* pk *as* $(g_1, g_2, g_3, h_1, h_2, \{f_{i,1}, f_{i,2}\}_{i=0}^{5})$.
- *pick* $\{\gamma_i\}_{i=1}^{5}$, $\{\delta_i\}_{i=1}^{2}$, r_1, *and* s_1 *at random and compute*

$$\bar{u}_1' = g^{\gamma_1} \cdot g_1^{r_1}, \quad \bar{u}_2' = g^{\gamma_2} \cdot g_2^{s_1}, \quad \bar{u}_3' = g^{\gamma_3} \cdot g_3^{r_1+s_1},$$
$$\bar{u}_4' = g^{\gamma_4} \cdot g^{x_1} \cdot h_1^{r_1} h_2^{s_1}, \quad \bar{u}_5' = g^{\gamma_5} \cdot g^{l_1},$$
$$\bar{v}_1' = \hat{e}(g_1, g^{\delta_1}) \cdot \prod_{i=1} \hat{e}(f_{i,1}, g^{\gamma_i}), \quad \bar{v}_2' = \hat{e}(g_2, g^{\delta_2}) \cdot \prod_{i=1} \hat{e}(f_{i,2}, g^{\gamma_i}).$$

- *output* $msg_1 = (\bar{u}_1', \bar{u}_2', \bar{u}_3', \bar{u}_4', \bar{u}_5', \bar{v}_1', \bar{v}_2')$ *and* $aux_1 = (\{\gamma_i\}_{i=1}^{5}, \{\delta_i\}_{i=1}^{2}, r_1, s_1)$.

$(msg_2, aux_2) \leftarrow \mathsf{BlindEnc_2}(\mathsf{pk}, l_2, x_2, msg_1)$

- *parse* pk *as* $(g_1, g_2, g_3, h_1, h_2, \{f_{i,1}, f_{i,2}\}_{i=0}^{5})$ *and* msg_1 *as* $(\bar{u}_1', \bar{u}_2', \bar{u}_3', \bar{u}_4', \bar{u}_5', \bar{v}_1', \bar{v}_2')$.
- *pick* r_2 *and* s_2 *at random and compute*

$$\bar{u}_1 = \bar{u}_1' \cdot g_1^{r_2}, \quad \bar{u}_2 = \bar{u}_2' \cdot g_2^{s_2}, \quad \bar{u}_3 = \bar{u}_3' \cdot g_3^{r_2+s_2},$$
$$\bar{u}_4 = \bar{u}_4' \cdot g^{x_2} \cdot h_1^{r_2} h_2^{s_2}, \quad \bar{u}_5 = \bar{u}_5' \cdot g^{l_2},$$
$$\bar{v} = \left(\textstyle\prod_{i=0} \hat{e}(f_{i,1}, \bar{u}_i)/\bar{v}_1'\right)^{r_2} \cdot \left(\textstyle\prod_{i=0} \hat{e}(f_{i,2}, \bar{u}_i)/\bar{v}_2'\right)^{s_2},$$

  where $\bar{u}_0 = g$.
- *output* $msg_2 = (\bar{u}_1, \bar{u}_2, \bar{u}_3, \bar{u}_4, \bar{u}_5, \bar{v})$ *and* $aux_2 = (r_2, s_2)$.

$\mathsf{c} \leftarrow \mathsf{UnblindEnc}(\mathsf{pk}, msg_2, aux_1)$

- *parse* pk *as* $(g_1, g_2, g_3, h_1, h_2, \{f_{i,1}, f_{i,2}\}_{i=0}^{5})$, msg_2 *as* $(\bar{u}_1, \bar{u}_2, \bar{u}_3, \bar{u}_4, \bar{u}_5, \bar{v})$ *and* $aux_1 = (\{\gamma_i\}_{i=1}^{5}, \{\delta_i\}_{i=1}^{2}, r_1, s_1)$.
- *compute*

$$u_1 = \bar{u}_1/g^{\gamma_1} = g_1^r, \quad u_2 = \bar{u}_2/g^{\gamma_2} = g_2^s, \quad u_3 = \bar{u}_3/g^{\gamma_3} = g_3^{r+s},$$
$$u_4 = \bar{u}_4/g^{\gamma_4} = g^{x_1+x_2} \cdot h_1^r h_2^s, \quad u_5 = \bar{u}_5/g^{\gamma_5} = g^{l_1+l_2},$$
$$v = \bar{v} \cdot \hat{e}(u_1 g_1^{-r_1}, g^{\delta_1}) \cdot \hat{e}(u_2 g_2^{-s_1}, g^{\delta_2}) \cdot \textstyle\prod_{i=0} \hat{e}(f_{i,1}^{r_1} f_{i,2}^{s_1}, u_i),$$

  where $u_0 = g$.
- *output* $\mathsf{c} = (u_1, u_2, u_3, u_4, v)$ *encrypted with label* u_5.

Correctness. Recall the structure of the ciphertext of the public-key encryption scheme described in Section 2: for a public key $\mathsf{pk} = (g_1, g_2, g_3, h_1, h_2, \{f_{i,1}, f_{i,2}\}_{i=0})$, label u_5, and randomly chosen $r, s \leftarrow \mathbb{Z}_q$, the ciphertext is computed as

$$(u_1, u_2, u_3, u_4, v) = \left(g_1^r,\ g_2^s,\ g_3^{r+s}, \quad m \cdot h_1^r h_2^s, \quad \prod_{i=0}^{5} \hat{e}(f_{i,1}^r f_{i,2}^s, u_i) \right), \text{where } u_0 = g.$$

Note that the protocol in Listing 3 computes a valid ciphertext because $u_1 = g_1^r$ for $r = r_1 + r_2$, $u_2 = g_2^s$ for $s = s_1 + s_2$, $u_3 = g_3^{r+s}$, $u_4 = m \cdot h_1^r h_2^s$ for $m = g^{x_1+x_2}$, and $v = \prod_{i=0} \hat{e}(f_{i,1}^r f_{i,2}^s, u_i)$. To see v is indeed computed this way, note that:

$$\bar{v} = \left(\prod_{i=0} \hat{e}(f_{i,1}, \bar{u}_i)/\bar{v}_1'\right)^{r_2} \cdot \left(\prod_{i=0} \hat{e}(f_{i,2}, \bar{u}_i)/\bar{v}_2'\right)^{s_2} = \frac{\prod_{i=0} \hat{e}(f_{i,1}^{r_2} f_{i,2}^{s_2}, u_i)}{\hat{e}(g_1, g^{\delta_1})^{r_2} \cdot \hat{e}(g_2, g^{\delta_2})^{s_2}}$$

and

$$\bar{v} \cdot \hat{e}\left(\frac{u_1}{g_1^{r_1}}, g^{\delta_1}\right) \cdot \hat{e}\left(\frac{u_2}{g_2^{s_1}}, g^{\delta_2}\right) = \bar{v} \cdot \hat{e}(g_1^{r_2}, g^{\delta_1}) \cdot \hat{e}(g_2^{s_2}, g^{\delta_2}) = \prod_{i=0} \hat{e}(f_{i,1}^{r_2} f_{i,2}^{s_2}, u_i).$$

Theorem 2. *The joint ciphertext computation protocol (Listing 3) strongly emulates the ideal two-party computation protocol (Listing 2) for function f_{JC}:* $\mathcal{P}_{\mathsf{jcc}}(\mathfrak{R}_1, \mathfrak{R}_2) \leq^{SS} \mathcal{I}_{\mathsf{tpc}}(f_{\mathsf{JC}}, \mathfrak{R}_1, \mathfrak{R}_2)$. We refer to the full paper for details.

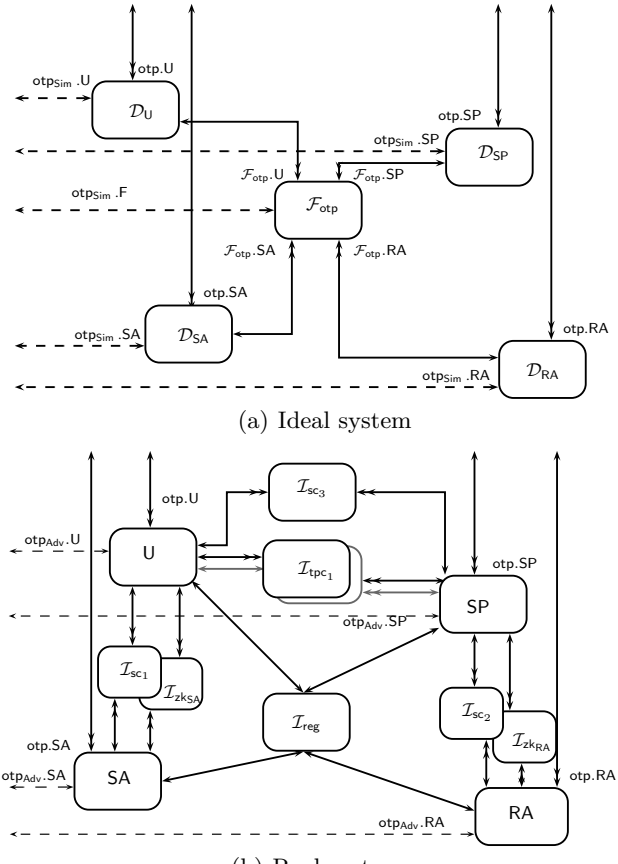
4 Oblivious Third Parties

Modeling oblivious third parties. Transactions in the real world can be intricately related. They may depend on many conditions, of which the verification can be deferred to a number of (as oblivious as possible) third parties. For the sake of concreteness, we now formally model a system that involves two oblivious third parties: a satisfaction authority and a revocation authority. In our example scenario, after a service enrollment between a user U and a service provider SP, the user ought to make a payment for the service before t_{due}. Upon request, the satisfaction authority SA checks that the user indeed made the payment and provides the user with a blinded transaction token. The user unblinds the token and publishes it to prove the satisfaction of the payment. Finally, the revocation authority RA reveals the user's identity to the service provider if no payment has been made before the payment deadline (i.e. no token corresponding to the enrollment was published).

We model the security and privacy requirements of such a system with the help of an ideal functionality $\mathcal{F}_{\mathsf{otp}}$. As usual, corruption is modeled via dummies D_{U}, D_{SP}, D_{SA}, D_{RA} that allow to access the functionality both over the environment interface (before corruption) and the network interface (after corruption).

The ideal system $\mathcal{I}_{\mathsf{otp}}$ is depicted in Figure 1(a) and consists of the ideal functionality connected to the dummy parties over delayed communication tapes. Listing 5 specifies the reactive behavior of $\mathcal{F}_{\mathsf{otp}}$. A user that can prove his identity with the help of a witness such that $(inst, (id, wit)) \in \mathfrak{R}$, is allowed to enroll. In particular, this interface supports the case where *wit* and *inst* are the secrets and the public key of a CL-signature [15] on the user's identity, i.e., an anonymous credential [14,5], or the opening and a commitment to the user's identity, i.e., a pseudonym [14]. For all these cases, the relation $\mathfrak{R}$ is tractable.

Enrollment consists of three rounds. The first round commits the user to her identity. The second round provides the user with a random satisfaction label



(a) Ideal system

(b) Real system

Fig. 1. The ideal OTP system $\mathcal{I}_{\mathsf{otp}}$ and its realization as a protocol $\mathcal{P}_{\mathsf{otp}}$: The realization makes use of ideal resources $\mathcal{I}_{\mathsf{sc}_i}$, $\mathcal{I}_{\mathsf{zk}_\mathsf{R}}$, $\mathcal{I}_{\mathsf{reg}}$, $\mathcal{I}_{\mathsf{jcc}_i}$ for secure communication, proofs of knowledge, key registration, and joint ciphertext computation respectively.

with respect to which she can satisfy the condition, e.g., make the necessary payment. In this round the user is also made aware of the due date t_{due} for the payment. Note that the user has to check that t_{due} fulfills reasonable uniformity constraints to protect her privacy. The last round gives the service provider the possibility to ask the identity revocation authority for the user's identity. As a common limitation with other escrow mechanisms for anonymous credentials, we cannot extract the identity itself, but only the image of a bijection of it. We model this by giving the simulator the possibility to choose the bijection. As the identity space of realistic systems is small enough to allow for exhaustive search, this is not a serious limitation.

The client interface towards the ideal oblivious parties, i.e., the interface of the user and the service provider respectively, consists of two messages `ReqAction`

and TestAction, with Action $\in$ {Satisfy, Open}. The obliviousness requirement guarantees that oblivious parties do not learn anything about the transactions of their clients. Indeed the decision of an oblivious party cannot be influenced in a transaction specific way, even if the other transaction participant colludes with the oblivious party. This is modeled with the help of test requests that are not related to any transaction. As these requests are indistinguishable from real requests, they allow the user to check whether the oblivious party indeed operates as required.[1]

Consequently, the decision of an oblivious party can only depend on explicit and relevant information. For *satisfaction*, this is the user known satisfaction label L with respect to which she makes her payment. For the *opening*, it is the transaction token T that is secret until after satisfaction, when it is learned by the user. We abstract from the way through which users make T available to the revocation authority, but envision some kind of anonymous publicly available bulletin board. It is in the responsibility of the user to make the token, learned during satisfaction, available to RA, and in the responsibility of RA to check its existence. All the protocol guarantees is that RA learns the same T value during *opening* as the user learned during *satisfaction*.

Listing 5: Functionality $\mathcal{F}_{\mathsf{otp}}$

Upon initialization, let state = "ready", $L = T = id = \widehat{T} = \widehat{id} = F = \mathbb{T} = \mathbb{L} = \epsilon$.
On $(\mathtt{SetF}, F', \mathbb{T}', \mathbb{L}')$ *from* $\mathsf{otp}_{\mathsf{Sim}}.\mathsf{F}$ *where state = "ready":*

- *abort if* F' *is not an efficient bijection or* $\mathbb{T}'$ *or* $\mathbb{L}'$ *are not of sufficient size; set* $F = F'$, $\mathbb{T} = \mathbb{T}'$, *and* $\mathbb{L} = \mathbb{L}'$.

On $(\mathtt{EnrollU}, inst, (id', wit'))$ *from* $\mathcal{F}_{\mathsf{otp}}.\mathsf{U}$ *where state = "ready":*

- *if* $(inst, (id', wit')) \notin \mathfrak{R})$ *abort;*
- *set state = "enrollu"; set id =* id'*; send* $(\mathtt{EnrollU}, inst)$ *to* $\mathcal{F}_{\mathsf{T}}.\mathsf{SP}$.

On $(\mathtt{DeliverEnrollU}, t_{due}')$ *from* $\mathcal{F}_{\mathsf{T}}.\mathsf{SP}$ *where state = "enrollu":*

- *set* $t_{due} = t_{due}'$*; set* T, L *to random values from* $\mathbb{T}$ *and* $\mathbb{L}$ *respectively;*
- *set state = "deliverenrollu"; send* $(\mathtt{DeliverEnrollU}, L, t_{due})$ *to* $\mathcal{F}_{\mathsf{otp}}.\mathsf{U}$.

On $(\mathtt{DeliverEnrollSP})$ *from* $\mathcal{F}_{\mathsf{otp}}.\mathsf{U}$ *where state = "deliverenrollu":*

- *set state = "enrolled"; send (*DeliverEnrollSP*) to* $\mathcal{F}_{\mathsf{otp}}.\mathsf{SP}$.

On $(\mathtt{ReqSatisfy})$ *from* $\mathcal{F}_{\mathsf{otp}}.\mathsf{U}$ *where* $L \neq \epsilon$ *and* $\widehat{T} = \epsilon$*:*

- *set* $\widehat{T} = T$*; send* $(\mathtt{ReqSatisfy}, L)$ *to* $\mathcal{F}_{\mathsf{otp}}.\mathsf{SA}$.

On $(\mathtt{TestSatisfy}, L', T')$ *from* $\mathcal{F}_{\mathsf{otp}}.\mathsf{U}$ *where* $\widehat{T} = \epsilon$*:*

- *set* $\widehat{T} = T'$*; send* $(\mathtt{ReqSatisfy}, L')$ *to* $\mathcal{F}_{\mathsf{otp}}.\mathsf{SA}$.

On $(\mathtt{Satisfy}, satisfied)$ *from* $\mathcal{F}_{\mathsf{otp}}.\mathsf{SA}$ *where* $\widehat{T} \neq \epsilon$*:*

[1] An extension that allows not only the requester, but arbitrary external parties, e.g. an auditor, to make test requests is a useful and cryptographically straightforward extension to this interface.

– *if satisfied, set* $m = (\texttt{Satisfy}, \widehat{T})$*, otherwise set* $m = (\texttt{Satisfy}, \bot)$*; set* $\widehat{T} = \epsilon$*; send* m *to* $\mathcal{F}_{\mathsf{otp}}.\mathsf{U}$.

On $(\texttt{ReqOpen})$ *from* $\mathcal{F}_{\mathsf{otp}}.\mathsf{SP}$ *where state = "enrolled" and* $\widehat{id} = \epsilon$*:*

– *set* $\widehat{id} = id$*; send* $(\texttt{ReqOpen}, T, t_{due})$ *to* $\mathcal{F}_{\mathsf{otp}}.\mathsf{RA}$.

On $(\texttt{TestOpen}, T', id', t_{due}{}')$ *from* $\mathcal{F}_{\mathsf{otp}}.\mathsf{SP}$ *where* $\widehat{id} = \epsilon$*:*

– *set* $\widehat{id} = id'$*; send* $(\texttt{ReqOpen}, T', t_{due}{}')$ *to* $\mathcal{F}_{\mathsf{otp}}.\mathsf{RA}$.

On $(\texttt{Open}, open)$ *from* $\mathcal{F}_{\mathsf{otp}}.\mathsf{RA}$ *where* $\widehat{id} \neq \epsilon$*:*

– *if open, set* $m = (\texttt{Open}, F(\widehat{id}))$*, otherwise set* $m = (\texttt{Open}, \bot)$*; set* $\widehat{id} = \epsilon$*; send* m *to* $\mathcal{F}_{\mathsf{otp}}.\mathsf{SP}$.

Implementing oblivious third parties. To construct a protocol that securely emulates the above functionality we make essential use of (adaptive chosen-ciphertext attack secure) encryption. As depicted in Figure 1(b) the protocol makes use of several cryptographic building blocks. But at the core of the protocol are two joint-ciphertext computations that, as described in Section 3, can be efficiently realized thanks to structure preserving encryption.

The enrollment protocol has a few more communication rounds, because of the zero-knowledge proofs, but otherwise closely follows the three phases of the ideal system. In the first phase the user commits to and proves her identity. Both the user and the service provider commit to randomness that they will use to jointly compute the transaction token T. The user proves knowledge of the opening of her commitment as part of the joint computation of the satisfaction ciphertext $\mathfrak{c}_1 = \mathsf{Enc}(pk_{\mathsf{SA}}, L, T \cdot g^r)$. In the second phase, the service provider transfers t_{due}, completes the joint ciphertext computation, and starts the computation of the revocation ciphertext $\mathfrak{c}_2 = \mathsf{Enc}(pk_{\mathsf{RA}}, g^{t_{due}}, (g^{id+r'}, T))$. In both cases, he proves knowledge of the opening to his commitment to guarantee that the transaction token is embedded correctly into both ciphertexts. The user outputs the label of $\mathfrak{c}_1$ as the random satisfaction label L. In the last phase the user again proves knowledge of openings for her commitments in the computation of $\mathfrak{c}_2$ to guarantee that it contains the transaction token T and a blinded user identity g^{id} under label $g^{t_{due}}$.

To satisfy her financial obligations, the user makes a payment with respect to label L and then asks the satisfaction authority to decrypt $\mathfrak{c}_1$. The user receives the blinded transaction token, that she unblinds using her locally stored randomness to learn T. She makes T available to the revocation authority, through some out-of-band anonymous bulletin board mechanism. Test satisfaction requests are just encryptions of blinded T' under label L'. To request the opening of a user identity, the service provider sends the ciphertext $\mathfrak{c}_2$ to the revocation authority, which checks the label t_{due}, decrypts the ciphertext to learn T and verifies whether T was posted by the user. If not, the revocation authority returns the blinded identity $g^{id+r'}$ to the service provider, which can unblind the identity. Test opening requests are just encryptions of T' and blinded $g^{id'}$ under label $t_{due}{}'$.

The Real System $\mathcal{P}_{\mathsf{otp}}$. We omit the details of the protocol and refer to the full version for the description of $\mathcal{P}_{\mathsf{otp}}$ and the proof that it securely emulates $\mathcal{F}_{\mathsf{otp}}$.

5 Conclusion

We propose the first public key encryption scheme that is structure preserving and secure against adaptive chosen ciphertext attacks. We demonstrate the usefulness of this new primitive by the joint ciphertext computation protocol and our proposal for instantiating oblivious third parties. We conjecture, however, that the combination of the structure preserving encryption scheme and efficient zero-knowledge proofs facilitate a much larger set of efficient protocol constructions. All protocols are proven secure in the universal composability model.

Acknowledgements. Jorn Lapon is supported by the Interuniversity Attraction Poles Programme Belgian State, Belgian Science Policy and the Research Fund K.U.Leuven, and the IWT-SBO projects DiCoMas and MobCom.

References

1. Abe, M., Fuchsbauer, G., Groth, J., Haralambiev, K., Ohkubo, M.: Structure-Preserving Signatures and Commitments to Group Elements. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 209–236. Springer, Heidelberg (2010)
2. Abe, M., Haralambiev, K., Ohkubo, M.: Signing on group elements for modular protocol designs. Cryptology ePrint Archive, Report 2010/133 (2010), `http://eprint.iacr.org`
3. Asokan, N., Shoup, V., Waidner, M.: Optimistic fair exchange of digital signatures. IEEE Journal on Selected Areas in Communications 18(4), 591–610 (2000)
4. Backes, M., Pfitzmann, B., Waidner, M.: The reactive simulatability (rsim) framework for asynchronous systems. Inf. Comput. 205(12), 1685–1720 (2007)
5. Belenkiy, M., Chase, M., Kohlweiss, M., Lysyanskaya, A.: P-signatures and Noninteractive Anonymous Credentials. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 356–374. Springer, Heidelberg (2008)
6. Boneh, D., Boyen, X., Shacham, H.: Short Group Signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
7. Camenisch, J., Shoup, V.: Practical Verifiable Encryption and Decryption of Discrete Logarithms. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 126–144. Springer, Heidelberg (2003)
8. Camenisch, J.: Group Signature Schemes and Payment Systems Based on the Discrete Logarithm Problem. Ph.D. thesis, ETH Zürich (1998)
9. Camenisch, J., Casati, N., Groß, T., Shoup, V.: Credential Authenticated Identification and Key Exchange. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 255–276. Springer, Heidelberg (2010)
10. Camenisch, J., Groß, T., Heydt-Benjamin, T.S.: Rethinking accountable privacy supporting services: extended abstract. In: ACM DIM – Digital Identity Management, pp. 1–8 (2008)

11. Camenisch, J., Haralambiev, K., Kohlweiss, M., Lapon, J., Naessens, V.: Structure preserving CCA secure encryption and its application to oblivious third parties. Cryptology ePrint Archive, Report 2011/319 (2011), http://eprint.iacr.org/
12. Camenisch, J., Kiayias, A., Yung, M.: On the Portability of Generalized Schnorr Proofs. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 425–442. Springer, Heidelberg (2009)
13. Camenisch, J., Krenn, S., Shoup, V.: A framework for practical universally composable zero-knowledge protocols. Cryptology ePrint Archive, Report 2011/228 (2011), http://eprint.iacr.org/
14. Camenisch, J., Lysyanskaya, A.: An Efficient System for Non-Transferable Anonymous Credentials with Optional Anonymity Revocation. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 93–118. Springer, Heidelberg (2001)
15. Camenisch, J., Lysyanskaya, A.: A Signature Scheme with Efficient Protocols. In: Cimato, S., Galdi, C., Persiano, G. (eds.) SCN 2002. LNCS, vol. 2576, pp. 268–289. Springer, Heidelberg (2003)
16. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: FOCS, pp. 136–145 (2001)
17. Cramer, R., Shoup, V.: A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (1998)
18. Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM Journal on Computing 33, 167–226 (2001)
19. Cramer, R., Shoup, V.: Universal Hash Rroofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002)
20. Dodis, Y., Haralambiev, K., López-Alt, A., Wichs, D.: Efficient Public-Key Cryptography in the Presence of Key Leakage. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 613–631. Springer, Heidelberg (2010)
21. Groth, J., Sahai, A.: Efficient Non-Interactive Proof Systems for Bilinear Groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008)
22. Katz, J., Vaikuntanathan, V.: Signature Schemes with Bounded Leakage Resilience. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 703–720. Springer, Heidelberg (2009)
23. Küsters, R.: Simulation-based security with inexhaustible interactive turing machines. In: 19th IEEE Computer Security Foundations Workshop, pp. 309–320. IEEE (2006)
24. Shacham, H.: A cramer-shoup encryption scheme from the linear assumption and from progressively weaker linear variants. Cryptology ePrint Archive, Report 2007/074 (2007), http://eprint.iacr.org